Pontificia Universidad Javeriana

Facultad de ciencias

Departamento de Matemáticas

Diana Carolina Gutierrez

Trabajo de grado para optar al titulo de maestría en Matemáticas

Dirigido por:

Camilo Ortiz Astorquiza, Ph.D
Ivan Contreras, Ph.D

Bogotá - Colombia
2021

Nota de advertencia

“ La universidad no se hace responsable por los conceptos emitidos por sus alumnos en sus trabajos de tesis. Solo velará por que no se publique nada contrario al dogma y a la moral católica y por que las tesis no contengan ataques personales contra persona alguna, antes bien se vea en ellas el anhelo de buscar la verdad y la justicia ”

Artículo 23 de la Resolución N° 13 de Julio de 1946

Submodularity and combinatorial representations for the multicommodity network design problem

Diana Carolina Gutierrez Diaz

Nota: Aprobado

Dirigido por:

Camilo Ortiz Astorquiza, Ph.D ______________________________

Ivan Contreras, Ph.D______________________________

Jurados:

Carlos Armando Zetina, Ph.D.______________________________

Rafael Santiago Gonzalez De Leon, Ph.D. ______________________________

Tesis presentada al Departamento de Matemáticas para optar al grado en Magister en Matemáticas

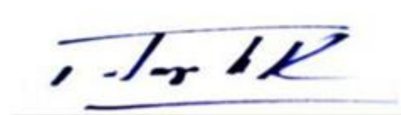
Pontificia Universidad Javeriana
Facultad de Ciencias
Departamento de Matemáticas
Bogotá - Colombia

Submodularity and combinatorial representations for the multicommodity network design problem

Diana Carolina Gutierrez Diaz

Aprobado:

Jhon Jairo Sutachan Rubio, Ph.D.
Director de Posgrado
Facultad de Ciencias



Alba Alicia Trespalacios Rangel, Ph.D.
Decana
Facultad de Ciencias

Pontificia Universidad Javeriana
Bogotá - Colombia
2022

Contents

1 Introduction **2**

2 Preliminaries **5**

2.1 Convex Analysis 5
2.2 Linear, Integer and Combinatorial Optimization 7
2.3 Solution methodologies 10
2.4 Complexity Theory 13
2.5 Submodularity 14
2.6 Matroids 16
2.6.1 Matroid Optimization 17

3 The multicommodity uncapacitated network design problem **19**

3.1 Problem definition 19
3.2 Literature Review 20

4 Combinatorial Representations for the MUND **23**

4.1 A natural combinatorial representation 23
4.2 An alternative representation 24
4.3 Matroid representation for the MUND 28

5 Worst-Case Bounds **32**

5.1 Greedy Heuristics for Special Cases of MUND 32
5.2 Worst-case Bound Results 35

6 Conclusion **41**

7 Bibliography **42**

Chapter 1

Introduction

Combinatorial optimization has its roots in combinatorics, operations research, and theoretical computer science. A great motivation for studying combinatorial optimization is that thousands of real-life problems can be formulated as abstract combinatorial optimization problems. Nonetheless, the mathematical concepts that lie behind the applications are interesting on their own and have been the focus of attention of many theoretical researchers for several decades now (see Korte et al. (2018); Nemhauser and Wolsey (1988);Wolsey (2020)).

In a combinatorial optimization problem one seeks to find the minimum (or maximum) value of a function over a domain which is defined as a set of combinatorial objects. Note that the set function that models the objective of the optimization problem is not unique. Moreover, it depends on the set of combinatorial objects used to model the original problem. This observation has proved to be very important when selecting the model and the corresponding solution approach since the computational efficiency can be significantly affected and the mathematical properties of one model versus another can have a tremendous impact when solving the model.

One particular class within combinatorial optimization problems is that referred to as Network Design Problems (NDPs) which are typically defined over a graph (or network). Broadly speaking, an NDP can be defined as follows. Given a graph, one must determine subsets of nodes or edges to activate/install (i.e., the design of the network) so that some requirement is satisfied while minimizing/maximizing an objective function usually associated with the costs/profits for opening or using the arcs/edges and the nodes. The type of requirement to be satisfied in the NDP provides a way of classifying them. For instance, NDPs are usually known as single or multicommodity NDPs depending on the characteristics of the requirements, that is, depending on whether only one type of flow in the network or several are sent. In multicommodity NDPs, a commodity is typically expressed as a pair of nodes origin-destination (OD).

NDPs lie at the heart of combinatorial optimization, turning into a major focus of attention for researchers and practitioners in many areas of knowledge (see Ahuja et al. (1993)), mainly because they constitute a rich area for several real applications while maintaining interesting modeling techniques and various solution methods developed for NDPs, those methods have been broadly used in different areas of Operations Research and combinatorial optimization.

Some particular cases of NDPs are well-known problems in various fields such as the shortest path problem (Nemhauser and Wolsey (1988), Ahuja et al. (1993)) which consist of finding a subset of edges that minimizes the total distance between pairs of nodes in the network; or the Traveling Salesman Problem which consists on determining, over a connected graph, a minimum cost Hamiltonian cycle in the network, (see Korte et al. (2018)).

From the applications perspective, NDPs may address strategical, tactical or operational decision-making situations. Moreover, because of the large number of practical problems that can be modeled through NDPs, this field has been constantly increasing over the past decades. As a result of the collective effort, a considerable amount of knowledge has been built, aiming to design and operate efficient systems in several sectors such as personnel scheduling (Balakrishnan and Wong (1990), Bartholdi et al. (1980)), service network design (Andersen et al. (2009), Crainic and Rousseau (1986)), logistics network design (Cordeau et al. (2006), Geoffrion and Graves (2010)), telecommunications and transportation planning (Melkote and Daskin (2001),Magnanti and Wong (1984)), physical networks, route networks and space-time networks (see Ahuja et al. (1993)).

On the other hand, NDPs are very challenging, in general they belong to the class of NP-hard problems, but rich problems in terms of mathematical structure. For instance, nowadays some of the most well-known solution methods in the field of Integer Programming were first studied for NDPs (see Conforti et al. (2014);Dionne and Florian (2006)). However, one of the most important properties in combinatorial optimization, e.g., submodularity (See Korte et al. (2018)), has thus far been elusive for the general case of NDPs. With this work we intend to fill this gap in the literature.

In this work we focus on one of the most well-known NDPs, the multicommodity uncapacitated network design problem (MUND). The MUND, broadly speaking, consists of selecting a set of arcs from a directed graph in such a way that all flow requirements are satisfied at a minimum total cost in the network; here, each flow requirement is associated to a commodity (pair OD) with a demand to be satisfied by delivering, from an origin to a destination node. The MUND generalizes a large class of well-known problems such as the traveling salesman problem, the uncapacitated lot-sizing problem, and the Steiner network design problem; see Ahuja et al. (1993),Nemhauser and Wolsey (1988) and Ortega and Wolsey (2003).

In particular, the contribution of this work is twofold. First, we establish a new combinatorial representation of the MUND whose objective function satisfies the submodularity property and second, based on such representation we propose worst-case bounds for a greedy heuristic that, to the best of our knowledge improve the current state-of-the-art results for particular cases of the MUND.

Therefore, this document is organized as follows. In Chapter 2 we present well-known results and the preliminaries needed throughout the rest of the document. In

Chapter 3 we formally define the MUND and provide a detailed literature review on the topic. In Chapter 4 we present a submodular representation and a matroid representation for the MUND. Then, in Chapter 5, using this submodular representation, we give a worse-case bound for a greedy algorithm for two special cases of the MUND problem and finally we state some conclusions and possible future works in chapter 6.

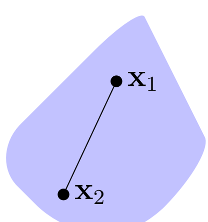
Chapter 2

Preliminaries

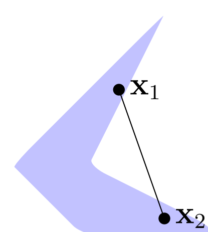
In this chapter we present some important definitions to understand and work with combinatorial optimization problems; those definitions and mathematical results come from convex analysis, linear programming, integer programming, combinatorial optimization, submodular optimization and complexity theory. The results presented in this section are derived from the article of Ortiz-Astorquiza et al. (2015) and the books of Bazaraa et al. (2011); Nemhauser and Wolsey (1988) and Ahuja et al. (1993).

2.1 Convex Analysis

Definition. *Convex sets. A set* X *in* $\mathbb{R}^n$ *is called a convex set if given any two points* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *in* X*, then* $\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 \in X$ *for each* $\lambda \in [0,1]$*. Any point of the form* $\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$ *where* $0 \leq \lambda \leq 1$ *is called a convex combination of* $\mathbf{x}_1$ *and* $\mathbf{x}_2$*.*



(a) Convex set



(b) Nonconvex set

Definition. *Extreme point. A point* x *in a convex set* X *is called an extreme point of* X *if* x *cannot be represented as a strict convex combination of two distinct points in* X*. In other words, if* $\mathbf{x} = \lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$ *with* $\lambda \in (0,1)$ *and* $\mathbf{x}_1, \mathbf{x}_2 \in X$*, then* $x = x_1 = x_2$*.*

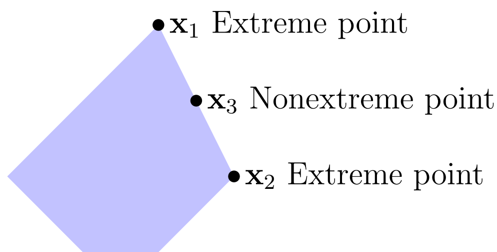


Figure 2.2: Example drawn on a convex set of extreme and nonextreme points.

Consider now a nonzero vector $\mathbf{p}$ in $\mathbb{R}^n$ and a scalar k. Then, we have the following definitions.

Definition. *A hyperplane in $\mathbb{R}^n$ generalizes the notion of a straight line in $\mathbb{R}^2$ and the notion of a plane in $\mathbb{R}^3$. A hyperplane H in $\mathbb{R}^n$ is a set of the form $\{\mathbf{x} : \mathbf{px} = k\}$. Here, $\mathbf{p}$ is called the normal or the gradient to the hyperplane.*

Definition. *A half-space is a collection of points of the form $\{\mathbf{x} : \mathbf{px} \geq k\}$. A half-space can also be represented as a set of points of the form $\{\mathbf{x} : \mathbf{px} \leq k\}$. The union of the two half-spaces $\{\mathbf{x} : \mathbf{px} \geq k\}$ and $\{\mathbf{x} : \mathbf{px} \leq k\}$ is $\mathbb{R}^n$.*

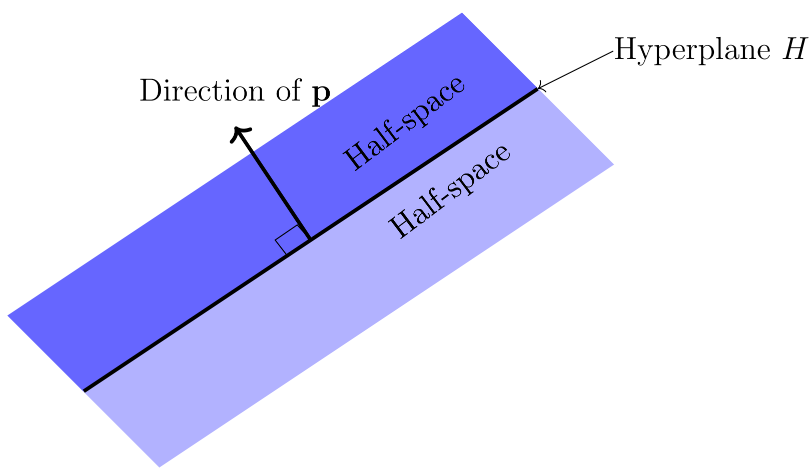


Figure 2.3: Half-spaces

Definition. *A polyhedral set (or polyhedron) is the intersection of a finite number of half-spaces.*

Also a polyhedron can be defined as a subset $P \subseteq \mathbb{R}^n$ described by a finite number of linear constraints $P = \{x \in \mathbb{R}^n : Ax \leq b\}$, with $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. The polyhedral set corresponding to the constraints in an optimization problem is called the feasible region and its elements are called feasible solutions i.e. the points satisfying all constraints. Finally, we present the formal definition of convex hull.

Definition. *Given a set $X \subseteq \mathbb{R}^n$, the convex hull of X, denoted* $\text{conv}(X)$ *is defined as:*

$$\text{conv}(X) = \left\{ x : x = \sum_{i=1}^{t} \lambda_i x^i, \sum_{i=1}^{t} \lambda_i = 1, \lambda_i \geq 0 \text{ for } i = 1, \cdots, t \text{ with } \left\{x^1, \cdots, x^t\right\} \subseteq X \right\}$$

In other words, the $\text{conv}(X)$ *is the (unique) minimal convex set containing X.*

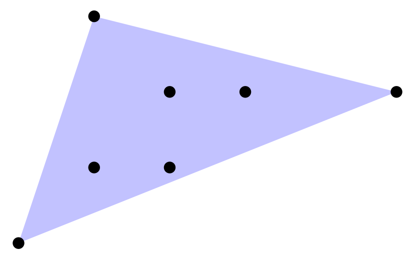


Figure 2.4: Example drawn of a convex hull

2.2 Linear, Integer and Combinatorial Optimization

Definition. *Consider a finite set $N = \{1, \cdots, n\}$ with corresponding weights c_j for each $j \in N$ and a set F of feasible subsets of N. The problem of finding a minimum weight feasible subset is the combinatorial optimization problem*

$$\min_{S \subseteq N} \left\{ \sum_{j \in S} c_j : S \in F \right\}$$

Given the discrete nature of combinatorial optimization problems, they are generally formulated as binary programs where all the variables must be $\{0, 1\}$ valued.

Consider the linear mixed integer programming problem defined as follows.

Definition. *If some but not all variables are integer, we have the (linear) mixed integer program (MIP)*

$$\min \left\{ cx + dy : Ax + By \leq b, x \in \mathbb{R}^n_+ \text{ and } y \in \mathbb{Z}^p_+ \right\}$$

where $x = (x_1, \cdots, x_n)$ is a vector of real variables and $y = (y_1, \cdots, y_p)$ is a vector of integer variables. An instance of MIP is specified by the data (c, d, A, B, b). The feasible region of MIP *is given by the set $S = \left\{(x, y) \in \mathbb{R}^n_+ \times \mathbb{Z}^p_+ : Ax + By \leq b\right\}$.*

The function $z = cx + dy$ is called the objective function and an optimal solution $z^* = (x^*, y^*)$ is a feasible point for which the objective value is minimum, i.e.,

$$cx^* + dy^* \leq cx + dy, \quad \forall (x, y) \in S$$

A special case of MIP is the following linear program.

Definition. *If all variables are in $\mathbb{R}_+$, we have the linear program (LP)*

$$\min \{cx : Ax \leq b \text{ and } x \in \mathbb{R}^p_+\}$$

Also, a linear programming problem can be described as the optimization problem of a linear objective function while satisfying a set of linear equality or inequality constraints. Classical techniques such as the simplex method as well as more sophisticated polynomial time algorithms, such as interior point methods, are nowadays capable of solving large-scale LP problems with millions of variables and constraints. When we restrict all the variables to be integer-valued, we have another special case of a MIP.

Definition. *If all variables are in $\mathbb{Z}_+$, we have the (linear) integer program (IP)*

$$\min \left\{cx : Ax \leq b \text{ and } x \in \mathbb{Z}^n_+\right\}$$

(Note that integer programming is a generalization of combinatorial problems).

In general, the solution of integer programming models is a challenging task. We cannot use classical machinery of convex optimization because $S = \left\{x \in \mathbb{Z}^n_+ : Ax \leq b\right\}$ is not a convex set and a linear function over $\mathbb{Z}^n_+$ is convex but non-differentiable. Therefore, we must resort to other mathematical techniques to prove that a particular solution is optimal by arguments other than convexity and differentiability.

When using integer programming, the first step is usually to represent the set of feasible solutions of an optimization problem with a polyhedron.

Definition. *A polyhedron $P \subseteq \mathbb{R}^{n+p}$ is a formulation for a set $X \subseteq \mathbb{Z}^n \times \mathbb{R}^p$ if and only if $X = P \cap (\mathbb{Z}^n \times \mathbb{R}^p)$.*

From the previous definition, it is clear that there is not a unique formulation for an optimization problem. This leads us to the following definition in order to be able to compare between the different formulations of a particular problem.

Definition. *Given a set $X \subseteq \mathbb{Z}^n$ and two formulations P_1 and P_2 for X then we say that P_1 is a better formulation than P_2, if $P_1 \subset P_2$.*

By this definition, we can conclude that no other formulation for a given set X is better than conv(X). Given that conv(X) is a polyhedral set, it can be represented by a finite set of linear constraints and thus, solved as a linear program. For some particular classes of combinatorial optimization problems the characterization of $\text{conv}(X)$ is known. However, for the vast majority of the problems that belong to the class NP-hard, the representation of $\text{conv}(X)$ remains unknown.

Several algorithms have been developed to solve MIPs. The key idea to these methods is usually to construct a sequence of lower bounds $\underline{z} \leq z^*$ and upper bounds $\bar{z} \geq z^*$ such that $z = \underline{z} = \bar{z}$. In practice, algorithms find an (not necessarily) increasing sequence of lower bounds and a (not necessarily) decreasing sequence of upper bounds, and stop when the difference between the lower bound and the upper bound is within a threshold value. We then need to find ways for obtaining such bounds. Following this idea, we must have in mind that in real applications we usually look for a balance between the time consumed by a model and its exactness. This means, if we can get a good approximation efficiently, sometimes is better than taking a long time to find the exact optimal solution. Obviously, this depends on the requirements of the problem and the particular application.

Many integer programming techniques use the simple idea of replacing a difficult MIP by an easier optimization problem whose optimal solution value is a lower bound for the MIP optimal solution value.

Definition. *A problem* $z_R = \min\{f(x) : x \in T \subseteq \mathbb{R}^n\}$ *is a relaxation of the integer program (IP)* $z = \min\{g(x) : x \in X \subseteq \mathbb{Z}^n\}$ *if the following two conditions are satisfied:*

- $X \subseteq T$
- $f(x) \leq g(x)$ *for all* $x \in X$.

An immediate result of this definition is that if (R) is a relaxation of (IP) then $z_R \leq z$. This means that any relaxation of the original problem will give us a lower bound if we are minimizing (an upper bound if we are maximizing). In the case of upper bounds when minimizing, every feasible solution is an upper bound and the problem lies in finding the smallest one. We present some methods to find upper bounds in later sections.

One of the most common relaxations of integer programs consists in dropping the integrality conditions.

Definition. *For an (IP)* $z = \min\{cx : x \in S\}$ *with* $S = X \cap \mathbb{Z}^n_+$ *the linear programming relaxation is given by* $z^{LP} = \min\{cx : x \in X\}$.

We present in the next proposition a known result that links the comparison between formulations and the LP relaxations.

Proposition 1. *Consider* P_1 *and* P_2 *two different formulations for an integer program and assume that* P_1 *is better than* P_2. *If* $z_i^{LP} = \min\{cx : x \in P_i\}$ *are the values of the associated linear programming relaxations, then* $z_1^{LP} \geq z_2^{LP}$.

2.3 Solution methodologies

The straightforward approach for solving a combinatorial optimization problem would be to list all feasible solutions and evaluate them in the objective function to select the one with minimum (or maximum) value. However, this brute force enumeration procedure is inefficient, so much so that even for relatively small instances of some problems computers can take a large amount of time, more than we are willing or able to wait. Consequently, several methods have been developed to find an optimal or near-optimal solution. Those methods that prove the optimality of the solution are known as exact solution techniques whereas those that find a feasible solution "hopefully" near the optimal value are usually categorized as (meta)heuristics.

Since most combinatorial optimization problems can be modeled via an IP or a MIP we mention some of the most important methods in the category of the exact solution techniques to solve IPs and MIPs, namely: the branch-and-bound method and the cutting planes algorithm. (see Land and Doig (2010), Gomory (1958)) In a branch-and-bound method, we would systematically partition the feasible region F into sub-regions $F^1, F^2, F^3, \ldots, F^K$. Let $\bar{x}$ denote the best feasible solution (in objective function value) we have obtained in prior computations. Suppose that for each $k = 1, 2, \ldots, K$, either F^k is empty or x^k is a solution of a relaxation of the set F^k and $c\bar{x} \leq cx^k$. Then no point in any of the regions F^1, $F^2, F^3, \ldots, F^k$ could have a better objective function value than $\bar{x}$, so $\bar{x}$ solves the original optimization problem. If $c\bar{x} > cx^k$, for any region F^k, we would need to subdivide this region by "branching" on some of the variables. Whenever we have satisfied the test $c\bar{x} \leq cx^k$ for all of the sub-regions (or we know they are empty), we have solved the original problem.

Proposition 2. *Let* $z = \min\{cx : x \in F\}$ *and* $F = F_1 \cup \cdots \cup F_k$ *be a decomposition of* F *into smaller sets, and let* $z^k = \min\{cx : x \in F_k\}$ *for* $k = 1, \cdots, K$*. Then,* $z = \min_k z^k$.

Now, in order to explain the cutting plane algorithm, we must have in mind that for a given (IP) $\min\{cx : x \in X\}$ where $X = \left\{x \in \mathbb{Z}^n_+ : Ax \leq b\right\}$ we can in theory, find the convex hull of X, i.e., $\mathrm{con}\, v(X) = \left\{x \in \mathbb{Z}^n_+ : \hat{A}x \leq \hat{b}\right\}$, which would lead us to simply solve its LP relaxation. However, finding $\mathrm{conv}(X)$ is not generally an easy (or efficient) task. What we can do in practice is to reduce the size of the set of feasible solutions by means of valid inequalities. Valid inequalities that are useful are the ones that are valid for X but are violated for the linear programming relaxation of (IP). To understand the concept we define the concepts of valid inequalities and separation problems.

Definition. *An inequality* $\alpha x \leq \alpha_0$ *is a valid inequality for* $X \subseteq \mathbb{R}^n$ *if* $\alpha x \leq \alpha_0$ *for all* $x \in X$.

Definition. *The cutting plane problem associated to a combinatorial optimization* *problem is the following: if given* $x^* \in \mathbb{R}^n$*, and* $x^* \notin \operatorname{con} v(X)$ *then find an inequality* $\alpha x \leq \alpha_0$ *satisfied by all points in* X *but violated by the point* x^*.

Suppose that $X = P \cap \mathbb{Z}^n$ with P a polyhedron and let V be a set of valid inequalities of X.

Result: Cutting plane algorithms
Given an initial polyhedron X. Set $i = 0$, $stop = false$, $P = P^0$.
while $stop = false$ **do**
  Solve the linear program $z^i = \min\{cx : x \in P^i\}$.
  if *The optimal solution* x^i *is integer* **then**
    $stop = true$
  else
    Solve the separation problem for x^i and P^i.
    if *There is an inequality that cuts off* x^i **then**
      $P^{i+1} = P^i \cap \{x : \alpha^i x \leq \alpha_0^i\}$
      $i = i + 1$
    else
      $stop = true$
    end
  end
end

The problem of finding the consecutive cutting planes that should be selected is difficult, and there is not an efficient way of generating the valid inequalities in order to get the convex hull of all the integer points contained in a given convex polyhedron P in "short" time.

Separately, when we are willing to accept a solution with a margin of error, say a "good solution" but not necessarily the optimal, usually the problem is to find solutions quickly. Then, we typically look for other methodologies to find feasible solutions for the IP or the MIP, known as heuristics and metaheuristics.

The most simple type of heuristics are the so-called greedy heuristics, this is an algorithm that always takes the best immediate, or local, solution while finding an answer (Black, 2015). For example, the greedy heuristic that we are going to use on this document follows the next steps: let $v(Q)$ be a real-valued function defined on all subsets of N and consider the problem $\max\{v(Q) : Q \subseteq N\}$. The idea of this greedy heuristic is: for a given set Q^t, select the element with the greatest immediate increase value, provided that such element exists, add it to the previous set $Q(t)$ and repeat until there is no more feasible elements with positive increase value.

Algorithm 1:

```
Result: Greedy Heuristic
Given an initial Q^0 = ∅. Set t = 1 and stop = false.
while stop = false do
    Let j_f = max_{j ∈ N\Q^{t-1}} v(Q^{t-1} ∪ {j}).
    if v(Q^{t-1} ∪ {j_t}) ⩽ v(Q^{t-1}) then
        Stop with Q^{t-1} a greedy solution.
        stop = true
    else
        if v(Q^{t-1} ∪ (j_t)) > v(Q^{t-1}), then
            set Q^t = Q^{t-1} ∪ {j_t},
            t + t + 1.
        else
            if Q^t = N then
                stop = true
            else
                t + t + 1
            end
        end
    end
end
```

Recall that (in general) we cannot expect that a greedy algorithm for a combinatorial problem yield an optimal solution. Others heuristics are the local search heuristic, the primal and dual heuristic (see Díaz et al. (2000), Nemhauser and Wolsey (1988)).

Definition. *An algorithm is a factor α approximation (α-approximation algorithm) for a problem if and only if for every instance of the problem it can find a solution within a factor α of the optimum solution.*

A α-approximation algorithm tries to find a feasible solution, as a heuristic, but the difference is that a α-approximation algorithm assures a feasible solution that is at most at a determinate distance from the optimal. This definition implies that the solution found by the algorithm is at most α times the optimum solution. If the problem is a maximization, $\alpha < 1$ and this definition guarantees that the approximate solution is at least α times the optimum.

Metaheuristics are special cases of heuristics, they refer to a master strategy that seeks to overcome local optimality, which is a usual problem of simple heuristics, and provide a general framework for the development of solution methods (see Gendreau et al. (2010)). One of the most used meta-heuristics is the greedy randomized adaptive search procedure (see Festa and Resende (2011)).

2.4 Complexity Theory

Complex theory seeks to analyze and classify problems according to how difficult it is to find an optimal solution. For each optimization problem, an associated (YES-NO)-instance decision problem, of the form: “Is there an $x \in S$ with value $cx \leq k$ for a given k ?” is used to define the class of legitimate problems.(An instance means a specific instance of that problem.). We are interested in complexity theory to be able to understand and classify the difficulty of the optimization problems, more specific of the muticommodity network design problem.

Definition (Big O notation)**.** *For two functions f, g defined on the natural numbers, $f(n)$ is order $g(n)$, written $f(n) = O(g(n))$, if there exists a constant C and a real number n_0 such that $|f(n)| \leq Cg(n)$ for all $n \geq n_0$.*

Definition (Running Time)**.** *The running time of algorithm A on an instance of size n is defined as $f_A(n) = \sup\{m : \exists$ input of size n such that it runs for m steps $\}$*

Definition (Polynomial time algorithm)**.** *An algorithm A is said to be a polynomial time algorithm for problem X if $f_A(n)$ is $O(n^p)$ for some fixed p.*

Definition ($\mathcal{NP}$)**.** *$\mathcal{NP}$ is the class of decision problems with the property that: for any instance for which the answer to the problem is yes, there is a polynomial time algorithm to proof the yes.*

Definition ($\mathcal{P}$)**.** *Let $\mathcal{P}$ be the class of problems that can be solved in polynomial time. Problem X is in $\mathcal{P}$ if and only if there is a polynomial time algorithm for solving X. A main theme of computational complexity is the question of inherent difference between problems known to be in $\mathcal{P}$ and those known to be in $\mathcal{NP}$. By definition we know $\mathcal{P} \subset \mathcal{NP}$ but to this date, there are problems in $\mathcal{NP}$ for which an algorithm running in polynomial time that can answer them has not been found yet. This does not imply that they do not exist nor that they do, so the question of $\mathcal{P} = \mathcal{NP}$ remains open.*

Definition. *If $Q, R \in NP$ and if an instance of Q can be converted in polynomial time to an instance of R, then Q is polynomially reducible to R.*

Definition. *The class of $\mathcal{NP}$-Complete problems, is the subset of problems $Q \in \mathcal{NP}$ such that for all $R \in \mathcal{NP}$, R is polynomially reducible to Q. An optimization problem for which its corresponding decision problem is $\mathcal{NP}$-Complete is called $\mathcal{NP}$-hard.*

The MUND problem, as some others network design problems, is know to be an $\mathcal{NP}$-hard problems, to see a proof you can see Johnson et al. (1978).

2.5 Submodularity

Submodularity is a fundamental property of set functions that arises in the context of combinatorial optimization. In general terms, it can be thought as the analogue of convexity for continuos functions, although in some cases it resembles concavity for certain properties (see Lovász (1983) Vondrák (2007)). In particular, the submodularity property has facilitated the study and development of algorithms for several classes of discrete optimization problems. For instance, the greedy heuristic with proven worst-case bounds on the optimal solution value for the general problem $\max_{\substack{S \subseteq E \\ |S| \leq k}} f(S)$ when E is a set and $f : 2^E \to \mathbb{R}_+$ is a nondecreasing submodular function. (see Nemhauser et al. (1978)). Below is the definition of submodularity for a function f.

Definition. *Let E be a finite set and f be a real-valued function defined on the set of subsets $\mathcal{E}$ and $\rho_e(S) = f(S \cup \{e\}) - f(S)$ be the incremental value of adding element e to the set S when evaluating the set function f.*

(a) f is submodular if $\rho_e(S) \geq \rho_e(T), \quad \forall S \subseteq T \subseteq E$ and $e \in E \backslash T$.
(b) f is nondecreasing if $\rho_e(S) \geq 0, \quad \forall S \subseteq E$ and $e \in E$.

Proposition 3. *(see Nemhauser et al. (1978). Proposition 2.1) Each of the following statements is equivalent and defines a submodular set function.*

(i) $z(A) + z(B) \geq z(A \cup B) + z(A \cap B), \quad \forall A, B \subseteq E$

(ii) $\rho_i(S) \geq \rho_j(T), \quad \forall S \subseteq T \subseteq E \quad$ *and* $\quad j \in E - T$

(iii) $\rho_j(S) \geq \rho_j(S \cup \{k\}), \quad \forall S \subseteq E \quad$ *and* $\quad j \in E - (S \cup \{k\})$

(iv) $$z(T) \leq z(S) + \sum_{j \in T-S} \rho_j(S) - \sum_{j \in S-T} \rho_j(S \cup T - \{j\}), \quad \forall S, T \subseteq E$$

(v) $$z(T) \leq z(S) + \sum_{j \in T-S} \rho_j(S), \quad \forall S \subseteq T \subseteq E$$

Proof. We will prove the equivalence of (i) and (ii), and then (iii) ⇒ (ii) ⇒ (iv) ⇒ (v) ⇒ (iii).
(i) ⇒ (ii). Take $S \subseteq T, j \notin T, A = S \cup \{j\}$ and $B = T$ in (i). This yields

$$z(S \cup \{j\}) + z(T) \geq z(T \cup \{j\}) + z(S)$$

or

$$\rho_j(S) = z(S \cup \{j\}) - z(S) \geq z(T \cup \{j\}) - z(T) = \rho_j(T)$$

(ii) $\Rightarrow$ (i). Let $\{j_1, \ldots, j_r\} = A - B$. From (ii) we obtain

$$\rho_{i_i}\left(A \cap B \cup \{j_1, \ldots, j_{i-1}\}\right) \geq \rho_h\left(B \cup \{j_1, \ldots, j_{i-1}\}\right), \quad i = 1, \ldots, r$$

Summing these r inequalities yields

$$z(A) - z(A \cap B) \geq z(A \cup B) - z(B)$$

(iii) $\Rightarrow$ (ii). Take $S \subseteq T, j \notin T$, and $T - S = \{j_1, \ldots, j_r\}$. Then from (iii) we have

$$\begin{aligned}&\rho_j(S) \geq \rho_j\left(S \cup \{j_1\}\right), \quad \rho_j\left(S \cup \{j_1\}\right) \geq \rho_j\left(S \cup \{j_1, j_2\}\right), \ldots, \\ &\rho_i\left(S \cup \{j_1, \ldots, j_{r-1}\}\right) \geq \rho_j(T)\end{aligned}$$

Summing these r inequalities yields (ii).
(ii) $\Rightarrow$ (iv). For arbitrary S and T with $T - S = \{j_1, \ldots, j_r\}$ and $S - T = \{k_1, \ldots, k_q\}$ we have

$$\begin{aligned} z(S \cup T) - z(S) &= \sum_{t=1}^{r}\left[z\left(S \cup \{j_1, \ldots, j_t\}\right) - z\left(S \cup \{j_1, \ldots, j_{t-1}\}\right)\right] \\ &= \sum_{t=1}^{r} \rho_{j_t}\left(S \cup \{j_1, \ldots, j_{t-1}\}\right) \leq \sum_{i=1}^{r} \rho_{j_t}(S) = \sum_{j \in T-S} \rho_j(S) \end{aligned} \tag{2.1}$$

where the inequality follows from (ii). Similarly

$$\begin{aligned} z(S \cup T) - z(T) &= \sum_{t=1}^{q}\left[z\left(T \cup \{k_1, \ldots, k_t\}\right) - z\left(T \cup \{k_1, \ldots, k_{t-1}\}\right)\right] \\ &= \sum_{t=1}^{a} \rho_{k_t}\left(T \cup \{k_1, \ldots, k_t\} - \{k_t\}\right) \geq \sum_{i=1}^{a} \rho_{k_t}\left(T \cup S - \{k_t\}\right) \\ &= \sum_{j \in S-T} \rho_j(S \cup T - \{j\}) \end{aligned} \tag{2.2}$$

We obtain (iv) by subtracting (2.2) from (2.1). (iv) $\Rightarrow$ (v). If $S \subseteq T, S - T = \emptyset$ and the last term of (iv) vanishes. (v) $\Rightarrow$ (iii). Substitute $T = S \cup \{j, k\}, j \notin S \cup \{k\}$ in (v) to obtain

$$z(S \cup \{j, k\}) \leq z(S) + \rho_j(S) + \rho_k(S)$$

or

$$\begin{aligned} \rho_j(S \cup \{k\}) &= z(S \cup \{j, k\}) - z(S \cup \{k\}) \\ &= z(S \cup \{j, k\}) - \rho_k(S) - z(S) \leq \rho_j(S). \end{aligned}$$

In many cases we consider nondecreasing submodular functions, which, in addition to (i), satisfy $z(S) \leq z(T), \forall S \subseteq T \subseteq E$ □

Proposition 4. *If z is a submodular set function on E with $-\theta \leq \rho_j(S)$, $\forall S \subseteq E, j \in E - S$, then*

$$z(T) \leq z(S) + \sum_{j \in T-S} \rho_j(S) + |S - T|\theta, \quad \forall S, T \subseteq E$$

Proof. Substitute the bounds on ρ_i into proposition 3, item (iv).

$$\begin{aligned} z(T) &\leq z(S) + \sum_{j \in T-S} \rho_j(S) - \sum_{j \in S-T} \rho_j(S \cup T - \{j\}), \quad \forall S, T \subseteq E \\ &\leq z(S) + \sum_{j \in T-S} \rho_j(S) - \sum_{j \in S-T} -\theta \\ &\leq z(S) + \sum_{j \in T-S} \rho_j(S) + |S - T|\theta \end{aligned}$$

□

2.6 Matroids

There are many equivalent ways to define a finite matroid, we present one that focuses on *independent sets.*

Definition. *A finite matroid M is defined as a pair $(E, \mathcal{I})$ where E is a finite set, called ground set, and $\mathcal{I}$ is a family of subsets of E, called the independent sets, such that $\mathcal{I}$ satisfies the following axioms:*

1. *The empty set is independent, i.e. $\emptyset \in \mathcal{I}$.*
2. *If $I_2 \subset I_1 \in \mathcal{I}$ then $I_2 \in \mathcal{I}$*
3. *If I_1 and I_2 are in $\mathcal{I}$, with $|I_2| > |I_1|$ then there exists $e \in I_2$ such that, $I_1 \cup e \in \mathcal{I}$*

Example.

One trivial example of a matroid $M = (E, \mathcal{I})$ is a uniform matroid in which

$$\mathcal{I} = \{X \subseteq E : |X| \leq k\}$$

for a given k. It is usually denoted as $U_{k,n}$ where $|E| = n$. A base is any set of cardinality k (unless $k > |E|$ in which case the only base is $|E|$). A free matroid is one in which all sets are independent; it is $U_{n,n}$.

Example.

A partition matroid, is a matroid in which E is partitioned into (disjoint) sets $E_1, E_2, \cdots, E_l$ and

$$\mathcal{I} = \{X \subseteq E : |X \cap E_i| \leq k_i \text{ for all } i = 1, \cdots, l\},$$

for some given parameters $k_1, \cdots, k_l$.

2.6.1 Matroid Optimization

Given a matroid $M = (E, \mathcal{I})$ and a objective function $c : E \to \mathbb{R}$, we are interested in finding an independent set S of M of maximum total value $c(S) = \sum_{e \in S} c(e)$. This is a fundamental problem. Note that the objective function c is also an additive function on the set E.

If all $c(e) \geq 0$, the problem is equivalent to finding a maximum value base in the matroid. If $c(e) < 0$ for some element e then, it will not be contained in any optimum solution, and thus we could eliminate such an element from the ground set. In the special case of a graphic matroid $M(G)$ defined on a connected graph G, the problem is thus equivalent to the maximum spanning tree problem which can be solved by a simple greedy algorithm.

The greedy algorithm we describe actually returns, for every k, a set S_k which maximizes $c(S)$ over all independent sets of size k. The overall optimum can thus simply be obtained by outputting the best of these. The greedy algorithm is the following:

Algorithm 2:

```
Result: Greedy algorithm for matroids
Sort the elements (and renumber them) such that ;
c(e_1) ≥ c(e_2) ≥ ··· ≥ c(e_{|M|})
Given an initial S_0 = ∅. Set t = 1 and stop = false.
while stop = false do
    if t ≤ |E| then
        if S_t + e_j ∈ I then
            t ← t + 1;
            S_t ← S_{t−1} + e_j;
            s_t ← e_j
        end
    else
        stop = true
    end
end
End with result the set S_t = {s_1, s_2, ..., s_t}
```

Proposition 5. *For any matroid $M = (E, \mathcal{I})$, the greedy algorithm above finds, for every t, an independent set S_t of maximum value among all independent sets of size t.*

Proof. We are going to give a contradiction proof. Suppose S_t is not the set of maximum value. Let $S_t = \{s_1, s_2, \cdots, s_t\}$ with $c(s_1) \geq c(s_2) \geq \cdots \geq c(s_t)$, and suppose T_t has greater value ($c(T_t) > c(S_t)$) where $T_t = \{t_1, t_2, \cdots, t_t\}$ with $c(t_1) \geq c(t_2) \geq \cdots \geq c(t_t)$. Let p be the first index such that $c(t_p) > c(s_p)$. Let $A =$

$\{t_1, t_2, \cdots, t_p\}$ and $B = \{s_1, s_2, \cdots, s_{p-1}\}$. Since $|A| > |B|$, there exists $t_i \notin B$ such that $B \cup \{t_i\} \in \mathcal{I}$. Since $c(t_i) \geq c(t_p) > c(s_p)$, t_i should have been selected when it was considered. To be more precise and detailed, when t_i was considered, the greedy algorithm checked whether t_i could be added to the current set at the time, say S. But since $S \subseteq B$, adding t_i to S should have resulted in an independent set since its addition to B results in an independent set. This gives the contradiction and completes the proof.

□

Chapter 3

The multicommodity uncapacitated network design problem

In this chapter we present a formal definition of the NDP under study, a brief literature review on the topic and special cases of the problem that are fundamental for the results presented in the subsequent chapters of this work. In particular, we focus our attention on the Multicommodity Uncapacitated Network Design Problem (MUND) which has shown to be central in the area of NDPs. For example, the MUND generalizes a large class of well-known problems such as the uncapacitated lot-sizing problem and the Steiner network design problem (Ortega and Wolsey (2003)).

3.1 Problem definition

We define the Multicommodity Uncapacitated Network Design problem (MUND) as follows. Let $G = (N, A)$ be a directed graph where N is the set of nodes and A is the set of arcs. Also, let K be a set of commodities where each commodity $k \in K$ is associated with a triplet $(O(k), D(k), d_k)$, where $O(k)$ is the origin node, $D(k)$ is the destination node and $d_k > 0$ is a demand to be served between $O(k)$ and $D(k)$. For each commodity k, we assume that there exists at least one path on G from $O(k)$ to $D(k)$ and we define the connected subnetwork $G^k = \left(N^k, A^k\right)$ that contains only nodes and arcs that belong to some path from $O(k)$ to $D(k)$. Also, consider f_{ij} to be the fixed costs for using an arc (i, j) linking node i to node j, c_{ij}^k be the costs per unit of commodity k delivered when using the arc (i, j). Hence, the most common definition of the MUND consists of selecting a subgraph of G, such that, all commodities are delivered from their origins to their destinations while minimizing the total cost, which is the sum of the variable cost and the set up cost (see Gendron et al. (1999) or Zetina et al. (2017)).

However, for the purpose of this research we consider the equivalent maximization counterpart of the problem. That is, given a revenue c^k gained per unit of commodity

k delivered, the maximization version of the MUND consist of selecting a subgraph of G, such that, all demands of commodities are satisfied while maximizing the total profit (consisting of the revenue minus the variable costs) minus the set up cost of arcs on the subgraph. The equivalence between the maximization and the minimization problems for the MUND follows if all demands are satisfied at optimality, then the total revenue is constant because the amount gained by deliver all commodities does not depend on the subgraph used to deliver them. Thus, the problem of maximizing a constant value minus the variable cost and the set up cost is equivalent to minimize the variable cost plus the set up cost, which implies the equivalence between the minimization and the maximization versions of the MUND.

Note that in this document we only consider the uncapacitated version of the problem. A natural extension that has also received important attention in the literature (see Gendron et al. (1999)) is to consider limited capacities in the nodes or the arcs to fulfill the demands of commodities. Nonetheless, we concentrate in the MUND since the problem already constitutes a major challenge and as explained in the following section it has been the focus of main research articles and books. More importantly, because, to the best of our knowledge, the results obtained in this thesis have not been published before.

3.2 Literature Review

The MUND has been broadly and deeply studied from different authors. One of the most important lines of research has been through Mixed Integer Programming (MIP). For example, a well-known MIP formulation for the MUND is the following (see Magnanti and Wong (1984))

$$\text{(P) minimize } \sum_{(i,j)\in A} f_{ij}y_{ij} + \sum_{k\in K}\sum_{(i,j)\in A} d_k c_{ij}^k x_{ij}^k$$

$$\text{subject to } \sum_{j\in N} x_{ji}^k - \sum_{j\in N} x_{ij}^k = \begin{cases} -1 & \text{if } i = O(k) \\ 1 & \text{if } i = D(k) \\ 0 & \text{otherwise} \end{cases} \forall i \in N, k \in K$$

$$x_{ij}^k \leq y_{ij} \quad \forall (i,j) \in A, k \in K$$

$$x_{ij}^k \geq 0 \quad \forall (i,j) \in A, k \in K$$

$$y_{ij} \in \{0,1\} \quad \forall (i,j) \in A,$$

where the binary variables y_{ij} are one if arc (i,j) is installed and zero otherwise and and the set of continuous variables x_{ij}^k represent the fraction of demand of commodity

k routed through arc (i, j). This type of formulation is usually solved by means of a general purpose solver which typically use, as their main solution technique, the Branch-and-Bound and Branch-and-Cut methods (see Nemhauser and Wolsey (1988) and Conforti et al. (2014)).

We are specially interested in two variants of the MUND problem. The first one is the restricted path cardinality constraint (MUND-RP) problem which is a special case of the MUND where a restriction on the total number of paths that can be chosen in the solution set is included. That is, the solution set is bounded by a natural number r. This problem has been studied before (see, for instance Bruglieri et al. (2006), they summarized the state of the art of research on combinatorial optimization problems with an exact cardinality constraint.)

The second variant is the MUND with a budget constraints (MUND-BC) problem where a maximum budget constraint on the total sum of the setup costs is included and all the set up cost are positive. The constraints of the MUND-BC have an impact in the way we can express the objective function, changing it for an objective function for which one no longer needs to consider the sum of the arcs' setup costs. Note that the constraint of the MUND-BC implies that the total number of arcs in the solution is limited to a budget, which directly implies that the number of arcs, and consequently the number of paths, are restricted for a maximum number r. The previous observation allows us to say that the MUND-BC can be seen like a MUND-RP. Some papers have studied the MUND-BC. e.g, Wong (1980) describes a polynomial time heuristic for the MUND-BC problem whose worst-case error ratio is two, considering all arcs set up cost equal to one.

There have been many studies about the MUND problem, some authors have approached this problem from a polyhedral perspective, i.e. incorporating valid inequalities (see Balakrishnan et al. (1991), Atamturk and Rajan (2002) splittable and unsplittable single arc-set relaxations, Raack et al. (2011) using graph connection properties and Chouman et al. (2017) inequalities related with the strong, cover, minimum cardinality, and others. Others authors have focused on the development of decomposition methods (Randazzo and Luna (2001), Frangioni and Gorgone (2014)) that exploit the problem structure to decompose the model into smaller subproblems using decomposition and relaxation methods like lagrangian relaxation, Benders decomposition and branch-and-bound or re-formulating the problem.

Another line of study has been the use of heuristics to obtain high quality solutions in short computing time, for the MUND problem the first proposed solution algorithm is an add-drop heuristic by Billheimer and Gray (1973). Other heuristics are those of Dionne and Florian (2006), Boffey and Hinxman (1979) and, some relevant heuristic results are given by Scott (1969), who proposed a greedy heuristic, which is known as a delete or backward algorithm. Gabrel et al. (2003), who modified the delete algorithm and proposed a link-rerouting and partial link-rerouting heuristics. Fragkos et al. (2017) who used Benders decomposition to solve a multi-period extension of the MUND, they experimented with the use of Pareto-optimal cuts and with the unified cut approach of Fischetti et al. (2010), obtaining significant computational gains. Also it is possible to find heuristics for the multicommodity capacitated network design problem (MCND), variant that generalizes the multicommodity uncapacitated network design problem (MUND), such as Kim and Pardalos (1999) (using dynamic slope scaling), Crainic et al. (2004) (using slope scaling and Lagrangean perturbations), Ghamlouche et al. (2003), Paraskevopoulos et al. (2016) (using scatter search with an iterated local search).

The main objective of this work is not to compare computational efficiency with other solution techniques but rather, provide a novel theoretical result that may yield new research directions and solution approaches for the general MUND. In particular, we introduce a greedy heuristic for variants of the MUND with a guaranteed worst-case bound, improving on the current state-of-the-art for the MUND problem. To achive that we give a combinatorial representation for the MUND whose set function satisfies the submodular property. This work is inspired by Nemhauser et al. (1978), who proves worst-case bounds for the uncapacitated location problem and Ortiz-Astorquiza et al. (2017) who proves the submodular property for the Multi-level uncapacitated facility location problem and proves worse-case bounds for the corresponding greedy heuristic.

Chapter 4

Combinatorial Representations for the MUND

As mentioned above the submodularity property plays a central role in the development of the results of this work. In this section we establish that the MUND can be modeled through different combinatorial representations, with different representations of the objective function, which in turn may or may not attain the submodularity property.

4.1 A natural combinatorial representation

Let $G = (N, A)$ be a graph following the definition on 3.1. Also, let p^k be a path with origin $O(k)$ and destination point $D(k)$ and c_{p^k} the profit obtained by delivering one unit of commodity k using path p^k. Define the set function $z : 2^A \to \mathbb{R}$ as:

$$z(S) = \sum_{k \in K} d^k \max_{p^k \in S} c_{p^k} - \sum_{(i,j) \in S} f_{ij} \tag{4.1}$$

Then, a natural way of expressing the MUND as a maximization problem over a set function is the following

$$\max_{S \subseteq A} \{z(S) | \ \exists \ p^k \in S \text{ for each } k \in K\}$$

Proposition 6. *The set function z defined in (4.1) is not a submodular function.*

Proof. Consider the MUND with a single commodity, fixed costs $f_{ij} = 0$, demand $d^1 = 1$, and additive profits as shown in the graph of Figure 4.1. where o is the origin node and d is the destination node. Then, define $R \subset S \subset A$ as

$$\begin{aligned} R &= \{(o, a), (a, b), (b, d)\} \\ S &= \{(o, a), (a, b), (b, d), (c, d)\} \end{aligned}$$

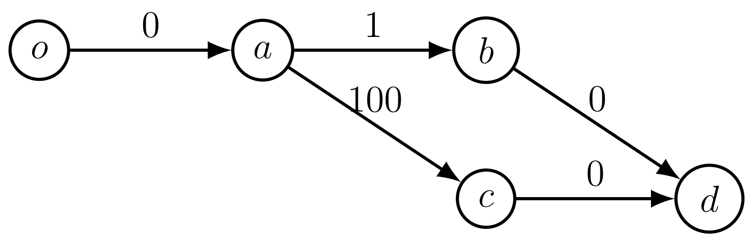


Figure 4.1: An instance of MUND

then $Z(R) = Z(S) = 1$, $Z(S \cup \{(a,c)\}) = 100$ and $Z(R \cup \{(a,c)\}) = 1$, then

$$Z(R \cup \{(a,c)\}) - Z(R) < Z(S \cup \{(a,c)\}) - Z(S)$$

thus, if $\rho_e(R) = Z(R \cup \{(a,c)\}) - Z(R)$ and $\rho_e(S) = Z(S \cup \{(a,c)\}) - Z(S)$ we conclude, due to $\rho_e(R) \leq \rho_e(S)$ with $R \subset S$, by using the definition of submodularity z is not a submodular set function. □

4.2 An alternative representation

Let $Q = Q_1 \cup Q_2 \cdots \cup Q_{|K|}$, be a set of simple paths in G, where Q_k is the set of all possible simple paths from $O(k)$ to $D(k)$, for $k \in K$ and let $W = Q \cup A$. Let f_{ij}, c_{ij}^k, c^k as defined in 3.1.

Additionally, for M a subset of W we define $S(M)$ to be the set of all paths in M, i.e.

$$S(M) = \{p | p \in M, p \text{ is a path}\}$$

and $S_k(M)$ to be the set of all paths from $O(k)$ to $D(k)$ that are in M, that is,

$$S_k(M) = \{p | p \in M, p \text{ is a path from } O(k) \text{ to } D(k)\}.$$

Then, $S(M) = S_1(M) \cup S_2(M) \cup \ldots S_{|K|}(M)$. Moreover, let $R(M)$ be the set of all arcs that are in M, i.e.,

$$R(M) = \{(i,j) \in A | (i,j) \in M\}.$$

In what follows, in order to simplify the notation, we refer to S_k, S and R when there is no ambiguity on the subset M that we are working on.

Now, let M be a subset of W, then we define:

$$
\begin{aligned}
f(M) &= -\sum_{(i,j)\in R} f_{ij} \\
h(M) &= \sum_{k\in K} d_k h^k(M) \\
h^k(M) &= \begin{cases} \max_{\mathbf{p}\in S_k}\left[c^k - c^k_{\mathbf{p}}\right] = \max_{\mathbf{p}\in S_k}\left[c^k - \sum_{(i,j)\in\mathbf{p}} c^k_{ij}\right] & if \qquad S_k \neq \emptyset \\ 0 & if \quad S_k = \emptyset \text{ and } M \neq \emptyset \end{cases} \\
h(\emptyset) &= \min_{\mathbf{p}\in Q}\left[c^k - c^k_{\mathbf{p}}\right] \\
z(M) &= h(M) + f(M) \\
&= \sum_{k\in K} d_k h^k(M) - \sum_{(i,j)\in R} f_{ij}
\end{aligned}
$$

and we define $f(\emptyset) = 0$.

Assumption 1. *We assume that the cost $c^k_{\boldsymbol{p}}$ from the use of a path $\boldsymbol{p}$ to deliver a unit of commodity k is additive. Then, for each $k \in K$ and $\boldsymbol{p}$ path from $O(k)$ to $D(k)$ we have*

$$c^k_{\boldsymbol{p}} = \sum_{(i,j)\in\boldsymbol{p}} c^k_{ij}$$

Then the MUND can be written as the problem of selecting a set $M \subseteq W$ such that $z(M)$ is maximum; this is,

$$\max_{M\subseteq W}\left\{z(M)|A(S(M)) = R(M), |S^k| > 0 \text{ for all } k \in K\right\}$$

where $A(S(M)) = \{(i,j) \in \mathbf{p} : \mathbf{p} \in S_k(M) \text{ for some } k \in K\}$.

Remark 1. *The MUND RP can be written as the MUND problem adding the cardinality constraint, thus*

$$\max_{M\subseteq W}\left\{z(M)|A(S(M)) = R(M), |S(M)| \leq r, |S^k| > 0 \textit{ for all } k \in K\right\} \tag{4.2}$$

Where r is the maximum number of paths on feasible solutions.

Remark 2. *For the MUND BC, we assume that $z(M) = h(M)$, then we can write the MUND BC as follows*

$$\max_{M \subseteq W} \left\{ z(M) | A(S(M)) = R(M), f(M) \leq b, |S^k| > 0 \text{ for all } k \in K \right\} \tag{4.3}$$

Where b is the maximum total sum of the setup costs.

A fundamental property of z is that of submodularity.

Proposition 7.

1. *$h(M)$ is submodular and nondecreasing.*
2. *$z(M) = h(M) + f(M)$ is submodular.*

Remark 3. *Proposition 7 is also true for the objective functions of problems (MUND RP) and (MUND BC). Note that for the MUND-BC, from proposition 7 we obtain that $z(M) = h(M)$ is a submodular and nondecreasing function.*

Proof. 1. Let $e \in W$, such that $e \in Q$. Then in order to proof the first statement of the proposition we set ${}^h\rho_e$

$$\begin{aligned} {}^h\rho_e(M) &= h(M \cup \{\mathbf{e}\}) - h(M) \\ &= \sum_{k \in K} d_k \left(h^k(M \cup \{\mathbf{e}\}) - h^k(M) \right) \\ &= \sum_{k \in K} d_k \left[\max_{\mathbf{p} \in S_k(M \cup \{e\})} \left[c^k - c^k_{\mathbf{p}} \right] - \max_{\mathbf{p} \in S_k(M)} \left[c^k - c^k_{\mathbf{p}} \right] \right] \\ &= \sum_{k \in K} d_k \max_{\mathbf{p} \in S_k(M \cup \{e\})} \left[c^k - \sum_{(i,j) \in \mathbf{p}} c^k_{ij} \right] - \sum_{k \in K} d_k \max_{\mathbf{p} \in S_k(M)} \left[c^k - \sum_{(i,j) \in \mathbf{p}} c^k_{ij} \right] \\ &= \sum_{k \in K} d_k \, \delta^e_k(M) \end{aligned}$$

Where we define δ_k as:

$$\delta^e_k(M) = \begin{cases} \left[c^k - c^k_e \right] - \max_{\mathbf{p} \in S_k(M)} \left[c^k - c^k_{\mathbf{p}} \right] & \text{if } c^k - c^k_e = \max_{\mathbf{p} \in S_k(M \cup \{e\})} \left[c^k - c^k_{\mathbf{p}} \right] \\ 0 & \text{other case.} \end{cases}$$

Note that if $e \in A$ then ${}^h\rho_e(M) = 0$ for all $M \in W$.

$${}^h\rho_e(\emptyset) = \left[c^k - c^k_e \right] - \min_{\mathbf{p} \in Q} \left[c^k - c^k_{\mathbf{p}} \right]$$

Note that $\delta_k^e(M) \geq 0$ for all $M \subset W$. Therefore, if $M_1 \subseteq M_2 \subseteq W$ and $\mathbf{e} \in \backslash M_2$ we want to deduce that ${}^h\rho_{\mathbf{e}}(M_1) \geq {}^h\rho_{\mathbf{e}}(M_2)$. Since the inequality is obtained if $M_1 = \emptyset$, we can assume M_1 to be non empty.

Case 1. If

$$\left[c^k - c_e^k\right] = \max_{\mathbf{p} \in S_k(M_2 \cup \{e\})} \left[c^k - c_{\mathbf{p}}^k\right] \tag{4.4}$$

due to $M_1 \subseteq M_2$ we get:

$$\left[c^k - c_e^k\right] = \max_{\mathbf{p} \in S_k(M_1 \cup \{e\})} \left[c^k - c_{\mathbf{p}}^k\right] \tag{4.5}$$

On the other hand, note

$$M_1 \quad \subseteq \quad M_2$$

then,

$$\max_{\mathbf{p} \in S_k(M_1)} \left[c^k - c_{\mathbf{p}}^k\right] \quad \leq \quad \max_{\mathbf{p} \in S_k(M_2)} \left[c^k - c_{\mathbf{p}}^k\right]$$

$$-\max_{\mathbf{p} \in S_k(M_1)} \left[c^k - c_{\mathbf{p}}^k\right] \geq -\max_{\mathbf{p} \in S_k(M_2)} \left[c^k - c_{\mathbf{p}}^k\right] \tag{4.6}$$

from 4.5 and 4.6 we get

$$\left[c^k - c_e^k\right] - \max_{\mathbf{p} \in S_k(M_1)} \left[c^k - c_{\mathbf{p}}^k\right] \geq \left[c^k - c_e^k\right] - \max_{\mathbf{p} \in S_k(M_2)} \left[c^k - c_{\mathbf{p}}^k\right] \tag{4.7}$$

and we get

$$\delta_k(M_1) \geq \delta_k(M_2).$$

Case 2. If

$$\left[c^k - c_e^k\right] \neq \max_{\mathbf{p} \in S_k(M_1 \cup \{e\})} \left[c^k - c_{\mathbf{p}}^k\right]$$

as $M_1 \subseteq M_2$ we get:

$$\left[c^k - c_e^k\right] \neq \max_{\mathbf{p} \in S_k(M_2 \cup \{e\})} \left[c^k - c_{\mathbf{p}}^k\right]$$

thus, by definition

$$\delta_k(M_1) = \delta_k(M_2) = 0$$

Case 3. If

$$\left[c^k - c_e^k\right] = \max_{\mathbf{p} \in S_k(M_1 \cup \{e\})} \left[c^k - c_{\mathbf{p}}^k\right]$$

and,

$$\left[c^k - c_e^k\right] \neq \max_{\mathbf{p} \in S_k(M_2 \cup \{e\})} \left[c^k - c_{\mathbf{p}}^k\right]$$

then by definition of δ_k we get

$$\delta_k(M_1) \geq 0 = \delta_k(M_2)$$

Therefore, from those three cases we have shown that $\delta_k(M_1) \geq \delta_k(M_2)$ for $M_1 \subseteq M_2$ and $k \in K$, hence:

$$\sum_{k\in K} d_k\ \delta_k(M_1) \geq \sum_{k\in K} d_k\ \delta_k(M_2) \geq 0$$

$$^{h}\rho_{\mathbf{e}}(M_1) \geq\ ^{h}\rho_{\mathbf{e}}(M_2) \geq 0$$

then h is submodular and nondecresing.

2. In order to prove the second statement we will proof that f is a submodular function.
Note that, for $e \in A \setminus R(M)$:

$$f(M \cup \{e\}) = -\sum_{(i,j)\in R(M\cup\{e\})} f_{ij} = -\sum_{(i,j)\in R(M)} f_{ij} - f_e$$

then

$$f(M \cup \{e\}) - f(M) = -f_e$$

thus for $M_1 \subseteq M_2 \subseteq W$ with $M_1 \neq \emptyset$ (if $M_1 = \emptyset$ we can replace and get easily the inequality) and $\mathbf{e} \in W \backslash M_2$ we get

$$\begin{aligned} -f_e &= -f_e \\ f(M_1 \cup \{e\}) - f(M_1) &= f(M_2 \cup \{e\}) - f(M_2) \\ ^{f}\rho_e(M_1) &=\ ^{f}\rho_e(M_2) \end{aligned}$$

which implies that f is submodular. Thus, z is the sum of two submodular function, then z is submodular.

□

4.3 Matroid representation for the MUND

Let $G = (N, A)$ be a directed graph for a MUND problem, the sets W,Q, A, the values f_{ij}, c_{ij}^k, c^k, and the functions $S(M), S_k(M), R(M), f(M), h(M), h^k(M), z(M)$ be defined as in 4.2.

Proposition 8. *Define $\mathcal{I}$ as the set family of subsets of W such that $I_1 \in \mathcal{I}$ if for each $k \in K$ there is at most one path from $O(k)$ to $D(k)$ in I_1, i.e. $|S_k(I_1)| \leq 1$ for all $k \in K$ and if p belongs to $S(I_1)$ then $^{z}\rho_p(I_1 \setminus \{p\}) > 0$. The pair $(W, \mathcal{I})$ have the following three properties:*

1. $\emptyset \in \mathcal{I}$

2. If $I_2 \subset I_1 \in \mathcal{I}$ then $I_2 \in \mathcal{I}$

3. If I_1 and I_2 are in $\mathcal{I}$, with $|I_2| > |I_1|$ then there exists $e \in I_2$ such that, $I_1 \cup e \in \mathcal{I}$

then the pair $(W, \mathcal{I})$ is a finite matroid.

Proof. 1. Note $\emptyset$ fills the requirements to belong to $\mathcal{I}$, then $\emptyset \in \mathcal{I}$.

2. If $I_2 \subset I_1 \in \mathcal{I}$ then I_1 does not have more than one path for each pair origin-destination, if $I_2 \subset I_1$ then I_2 does not have more than one pair either. Furthermore, if $p \in I_2 \subset I_1$ then ${}^z\rho_p(I_1 \setminus \{p\}) > 0$, due to z is a submodular function (proposition 3) then ${}^z\rho_p(I_2 \setminus \{p\}) \geq^z \rho_p(I_1 \setminus \{p\}) > 0$. Hence, we conclude $I_2 \in \mathcal{I}$.

3. If I_1 and I_2 are in $\mathcal{I}$, with $|I_2| > |I_1|$ then there exist an arc or a path such that $e \in I_2$ but $e \notin I_1$. In the case e is an arc then $I_1 \cup \{e\}$ still have no more than one path for each $k \in K$ and for all paths in I_1 we have

$$\begin{aligned} {}^z\rho_p(I_1 \cup \{e\} \setminus \{p\}) &= z(I_1 \cup \{e\}) - z(I_1 \cup \{e\} \setminus \{p\}) \\ &= (h(I_1 \cup \{e\}) + f(I_1 \cup \{e\})) - (h(I_1 \cup \{e\} \setminus \{p\}) + f(I_1 \cup \{e\} \setminus \{p\})) \end{aligned}$$

Note that $f(I_1 \cup \{e\} \setminus \{p\}) = f(I_1 \cup \{e\})$ and since e is an arc, then $h(I_1 \cup \{e\}) = h(I_1)$ and $h(I_1 \cup \{e\} \setminus \{p\}) = h(I_1 \setminus \{p\})$. Thus,

$$\begin{aligned} {}^z\rho_p(I_1 \cup \{e\} \setminus \{p\}) &= (h(I_1) + f(I_1 \cup \{e\})) - (h(I_1 \setminus \{p\}) + f(I_1 \cup \{e\})) \\ &= h(I_1) - (h(I_1 \setminus \{p\}) \\ &= h(I_1) + f(I_1) - (h(I_1 \setminus \{p\}) + f(I_1 \setminus \{p\})) \\ &=^z \rho_p(I_1 \setminus \{p\}) > 0. \end{aligned}$$

In the case e is a path, then $e \in I_2$ and $e \notin I_1$. Therefore there exist one pair origin-destination $(O(k*), D(k*))$ for some $k*$, such that I_1 does not have a path with those origin-destination. Then, applying the item 2 we have $\{p\} \in \mathcal{I}$ then

$${}^z\rho_p(\emptyset) = z(p) - z(\emptyset) = d_{k*}[c^{k*} - c_p^{k*}] - \sum_{(i,j) \in p} f_{ij} - [\min_{\mathbf{p} \in Q} c^k - c_{\mathbf{p}}^k] > 0$$

Moreover,

$$\begin{aligned} {}^z\rho_p(I_1 \cup e \setminus \{p\}) &= z(I_1 \cup e) - z(I_1 \cup e \setminus \{p\}) \\ &= d_{k*}[c^{k*} - c_p^{k*}] \\ &\geq d_{k*}[c^{k*} - c_p^{k*}] - \sum_{(i,j) \in p} f_{ij} \\ &\geq d_{k*}[c^{k*} - c_p^{k*}] - \sum_{(i,j) \in p} f_{ij} - [\min_{\mathbf{p} \in Q} c^k - c_{\mathbf{p}}^k] \\ &=^z \rho_p(\emptyset) > 0 \end{aligned} \tag{4.8}$$

Thus, we conclude $(W, \mathcal{I})$ is a matroid. ∎

Assumption 2. *The MUND problem can be written as the problem of selecting a set $I \in \mathcal{I}$ such that $z(I)$ is maximum; that is,*

$$\max_{I \subseteq \mathbb{I}} \left\{ z(I) | A(S(I)) = R(I), |S^k| > 0 \text{ for all } k \in K \right\}$$

Proposition 9. *Define $\mathcal{I}^*$ as the set family of subsets of Q such that $I_1 \in \mathcal{I}^*$ if for each $k \in K$ there is at most one path from $O(k)$ to $D(k)$ in I_1, i.e. $|S_k(I_1)| \leq 1$ for all $k \in K$. Then, for the pair $(Q, \mathcal{I}^*)$ we have*

1. *$\emptyset \in \mathcal{I}^*$*

2. *If $I_2 \subset I_1 \in \mathcal{I}^*$ then $I_2 \in \mathcal{I}^*$*

3. *If I_1 and I_2 are in $\mathcal{I}^*$, with $|I_2| > |I_1|$ then there exists $e \in I_2$ such that, $I_1 \cup \{e\} \in \mathcal{I}^*$*

then the pair $(Q, \mathcal{I}^*)$ is a finite matroid.

Proof. 1. Note $\emptyset$ fill the requirements to belong to $\mathcal{I}^*$, then $\emptyset \in \mathcal{I}^*$.

2. If $I_2 \subset I_1 \in \mathcal{I}^*$ then I_1 do not have more than one path for each $k \in K$. Since $I_2 \subset I_1$, then I_2 do not have more than one path for each $k \in K$ either, this imply $I_2 \in \mathcal{I}^*$.

3. If I_1 and I_2 are in $\mathcal{I}^*$, with $|I_2| > |I_1|$ then there is at least one commodity k' such that there is a path $p_{k'}$ for commodity k' in I_2 but there is not path for commodity k' in I_1; then, as $I_1 \in \mathcal{I}^*$ we know I_1 have at most one path for each commodity and it do not have path for commodity k', then the set $I_1 \cup \{p_{k'}\}$ have at most one path for each commodity, then $I_1 \cup \{p_{k'}\} \in \mathcal{I}^*$

thus, we conclude $(Q, \mathcal{I}^*)$ is a matroid. □

Assumption 3. *We define the set function $z^* : Q \to \mathbb{R}$ as:*

$$z^*(I) = z(I \cup A(I))$$

then, the MUND problem can be written as the problem of selecting a set $I \in \mathcal{I}^*$ such that $z^*(I)$ is maximum; that is,

$$\max_{I \subseteq \mathbb{I}^*} \left\{ z^*(I) : |S^k| > 0 \text{ for all } k \in K \right\}. \tag{4.9}$$

note that if there is at least one path for each commodity such that the revenue cost minus the costs paid for using and deliver the commodity for all the arcs on the paths is greater than zero then the requirement is not needed. This implies that, in this case, the MUND problem can be written as the problem of selecting a set $I \in \mathcal{I}^*$ such that $z^*(I)$ is maximum; that is,

$$\max_{I \subseteq \mathbb{I}^*} \left\{ z^*(I) \right\}. \tag{4.10}$$

Remark 4. *Note that we can not apply the proposition 5 to the matroid* $(Q, \mathcal{I}^*)$ *with objective function* z^* *because to apply this proposition we need the objective function to be an additive function (which is not* z^**) but if we consider the case the very specific case of the MUND where we omit the sum of the set up costs in the objective function, that is, we can take* $z^*(I) = h^*(I) = h(I)$ *and add a cardinality constraint on the number of paths. Then, since* h *is an additive function, we can use the greedy algorithm and the proposition 5 to get the optimal solution (we assume there is not paths with negative or zero values of the objective function on each iteration, in case that there are, we remove them from the set of available paths).*

Remark 5. *We can get the same results, the alternative representation, the proposition 7 and the matroid representation in the case that the graph* G *is an underacted graph; the difference relies on the set* W *of all possible paths, from a origin to a destination, on the graph.*

Chapter 5

Worst-Case Bounds

In this chapter we describe a greedy heuristic for the special cases of MUND, namely the MUND-RP and the MUND-BC. Moreover, we also present worst-case bounds for the corresponding greedy heuristics based on the results obtained in the previous chapters.

5.1 Greedy Heuristics for Special Cases of MUND

Let M^t denote the current solution at the iteration t, $\rho_A(M) = z(M \cup A) - z(M)$ be the incremental value of adding the subset A to the set M and ρ^t the maximum possible increment at the iteration t. We consider a heuristic that stops after a specific fixed number r and constructs a feasible solution by adding at each iteration a subset of elements of W satisfying the feasibility condition, i.e., $A(S(M)) = R(M)$. This is done by considering a candidate subset that contains exactly one path $q \in Q$ with its corresponding arcs $A(\{q\})$. Thus, we define $A_q = \{q\} \cup A(\{q\})$. For the following heuristics we suppose there exist at least one path on every Q_k such that $\rho_p(\emptyset) > 0$.

Algorithm 3: Greedy heuristic for the MUND-RP

Result: A greedy solution for the MUND
Let $M^0 \leftarrow \emptyset$,$W^0 \leftarrow W$ and $t \leftarrow 1$;
initialization;
while $t < r + 1$ *and* $W^{t-1} \neq \emptyset$ **do**
  Select $q^* \in W$ for which;
  $\rho_{A_{q^*}}(M^{t-1}) = \max_{q \in W^{t-1}} \rho_{A_q}(M^{t-1})$;
  Set $\rho^{t-1} \leftarrow \rho_{A_{q^*}}(M^{t-1})$;
  if $\rho^{t-1} \leq 0$ **then**
    Stop with M^{t-1} as the greedy solution ;
  else
    Set $M^t \leftarrow M^{t-1} \cup A_{q^*}$;
    $W^t \leftarrow W^{t-1} \setminus q^*$;
  end
  $t \leftarrow t + 1$;
end
Stop with M^{t-1} as the greedy solution;

Since the budget constraint directly restricts the arcs that can be opened and hence the paths that can be selected, the MUND-BC can be seen as a MUND-RP, then this algorithm can be used for both variants but it is necessary to have a stopping criterion accordingly with the specific case that we are working on to obtain the worst-case bound results that are presented later in this chapter.

For the MUND-RP we use the objective function specified in remark 2, and for MUND-BC we use the the objective function specified in remark 1, and the constant r also depends on the variant, for the MUND-RP r is the number of the restriction of the maximum number of path and, for the MUND-BC, the r is obtained by finding which is the maximum number of arcs that can be selected such that hold the minimum set up cost and the budget constraint.

We also consider an independent algorithm for the MUND-BC, let's define z as in remark 2 and b as the budget constraint, then

Algorithm 4: Greedy heuristic for the MUND-BC

Result: A greedy solution for the MUND-BC
Let $M^0 \leftarrow \emptyset$,$W^0 \leftarrow W$ and $t \leftarrow 1$;
initialization;
while $t < r+1$ *and* $W^{t-1} \neq \emptyset$ **do**
  Select $q^* \in W$ for which;
  $\rho_{A_{q^*}}(M^{t-1}) = \max\limits_{q \in W^{t-1}} \rho_{A_q}(M^{t-1})$;
  Set $\rho^{t-1} \leftarrow \rho_{A_{q^*}}(M^{t-1})$;
  if $\rho^{t-1} \leq 0$ *or* $f(M^{t-1} \cup A_q) > b$ **then**
    Stop with M^{t-1} as the greedy solution ;
  else
    Set $M^t \leftarrow M^{t-1} \cup A_{q^*}$;
    $W^t \leftarrow W^{t-1} \setminus q^*$;
  end
  $t \leftarrow t+1$;
end
Stop with M^{t-1} as the greedy solution;

The following results are valid for both algorithms.

Remark 6. *The greedy heuristic stops after* r *iterations for the MUND-RP or MUND-BC.*

Proposition 10. *The greedy heuristic for the MUND can be executed in* $O(r|A||K|(1+|N|))$ *time.*

Proof. At each iteration we have to identify the paths with maximum increment in the set W^{t-1}. First of all, we identify q^* such as $\rho_{A_{q^*}}(M^{t-1}) = \max\limits_{A_q \in W^{t-1}} \rho_q(M^{t-1})$, to do this consider the auxiliary graph $G^t = (N, Arc^t)$ where $Arc^t = R(M^t) \cup R(W^t)$ with $R(W^t)$ set of available arcs to enter in the solution set at the $t+1$ iteration and M^t solution set at t iteration. For each commodity $k \in K$ we define for each $w_{ij} \in Arc^t$ a length given by:

$$w_{ij} = \begin{cases} f_{ij} + c_{ij}^k & if \quad (ij) \notin R(M^t) \\ \\ c_{ij}^t & if \quad (ij) \in R(M^t). \end{cases}$$

This operation takes $O(|A||K|)$ and we compute a candidate path q_k solving the shortest path problem from $O(k)$ to $D(k)$, where k is not supplied yet, this can be done by using the Fifo *label- correcting* algorithm in $O(|K||N||A|)$ (Ahuja at 1993), then select the path q^* from $q^* = \max\limits_{k \in K} \rho_{A_{q_k}}(M^{t-1})$. Thus, each iteration of the greedy heuristic takes a total of $O(|A||K|(1+|N|))$, given that there are at most r iterations, then the greedy heuristics takes $O(r|A||K|(1+|N|))$. □

5.2 Worst-case Bound Results

Proposition 11. *For $M_1 \subseteq M_2 \subseteq W$ and any subset $A \subseteq W \setminus M_2$,*

$$\rho_A(M_1) \geq \rho_A(M_2).$$

Proof. The result follows directly from Proposition 7 □

Moreover, since the set W is finite and given the definition of the function z, there exists a $\theta \geq 0$ for which $\rho_A(M) \geq -\theta$ for $M \subseteq W$ and $A \subseteq W \backslash M$. In the case of having a nondecreasing set function (e.g. h) $\theta = 0$

Proposition 12. *For all $M_1, M_2 \subseteq W$ such that $A(S(M_1)) = R(M_1)$ and $A(S(M_2)) = R(M_2)$, let's denote $S_1 = S(M_1)$ and $S_2 = S(M_2)$,*

$$z(M_2) \leq z(M_1) + \sum_{q \in S_2 \setminus S_1} \rho_{A_q}(M_1) + |S_1 \setminus S_2|\theta$$

Proof. Let $M_1, M_2 \subseteq W$, with $|S_1 \backslash S_2| = \beta, |S_2 \backslash S_1| = \alpha$. Consider the sets A_q with $q \in S_2 \backslash S_1$ and similarly $B_s = A_s$ with $s \in S_1 \backslash S_2$, as defined before. Also, we enumerate the paths $q \in S_2 \backslash S_1$ as $q = 1, \ldots, \alpha$ and similarly those in $S_1 \backslash S_2$ as $s = 1, \ldots, \beta$. Then

$$z(M_1 \cup M_2) - z(M_1) \leq \sum_{q \in S_2 \setminus S_1} \rho_{A_q}(M_1), \tag{5.1}$$

since

$$\begin{aligned} & z(M_1 \cup M_2) - z(M_1) \\ & = z\left(M_1 \cup A_1\right) - z(M_1) + z\left(M_1 \cup A_1 \cup A_2\right) \\ & -z\left(M_1 \cup A_1\right) + \cdots + z\left(M_1 \cup A_1 \cup \cdots \cup A_\alpha\right) \\ & -z\left(M_1 \cup A_1 \cup \cdots \cup A_{\alpha-1}\right) \\ & = \sum_{i=1}^{\alpha} \rho_{A_i}\left((M_1) \cup A_1 \cup \cdots \cup A_{i-1}\right) \\ & \leq \sum_{i=1}^{\alpha} \rho_{A_i}(M_1) = \sum_{q \in S_2 \setminus S_1} \rho_{A_q}(M_1) \end{aligned}$$

where the inequiality follows from Proposition 11. Similary,

$$z(M_1 \cup M_2) - z(M_2) \geq \sum_{s \in S_1 \setminus S_2} \rho_{B_s}\left(M_2 \cup M_1 \backslash B_s\right), \tag{5.2}$$

since: $z(M_1 \cup M_2) - z(M_2)$

$$
\begin{aligned}
&= z\left(M_2 \cup B_1\right) - z(M_1) + z\left(M_2 \cup B_1 \cup B_2\right) \\
&- z\left(M_2 \cup B_1\right) + \cdots + z\left(M_2 \cup B_1 \cup \cdots \cup B_\beta\right) \\
&- z\left(M_2 \cup B_1 \cup \cdots \cup B_\beta\right) \\
&= \sum_{i=1}^{\beta} \rho_{B_i}\left((M_2) \cup B_1 \cup \cdots \cup B_{i-1}\right) \\
&\geq \sum_{i=1}^{\beta} \rho_{B_i}(M_1 \cup M_2 \setminus B_i) = \sum_{s \in S_1 \setminus S_2} \rho_{B_s}\left(M_2 \cup M_1 \setminus B_s\right)
\end{aligned}
$$

where the inequiality follows from Proposition 11. Substracting (5.1) from (5.2), we obtain

$$
z(M_2) \leq z(M_1) + \sum_{q \in S_2 \setminus S_1} \rho_{A_q}(M_1) - \sum_{s \in S_1 \setminus S_2} \rho_{B_s}\left(M_2 \cup M_1 \setminus B_s\right).
$$

Since $\rho \geq -\theta$, it follows that:

$$
z(M_2) \leq z(M_1) + \sum_{q \in S_2 \setminus S_1} \rho_{A_q}(M_1) + \beta\theta
$$

□

Let Z be the optimal solution value of an instance of the MUND and let Z^G be the value of a solution obtained using Algorithm 1. Thus, $Z^G = z(\varnothing) + \rho_0 + \rho_1 + \cdots + \rho_{t^*-1}$, with $t^* \leq r$.

Proposition 13. *If the greedy heuristic for the MUND stops after $t^* \leq r$ iterations then:*

$$
\mathrm{Z} \leq z(\varnothing) + \sum_{i=0}^{t-1} \rho_i + r\rho_t + t\theta \quad t = 0, \ldots, t^* - 1. \tag{5.3}
$$

and also:

$$
Z \leq z(\varnothing) + \sum_{i=0}^{t^*-1} \rho_i + t^*\theta \qquad \text{if } t^* < r. \tag{5.4}
$$

Proof. By Proposition 7 we have $z(M_2) \leq z(M_1) + \sum_{q \in M_2 \setminus M_1} \rho_q(M_1) + |M_1 \setminus M_2|\theta$.

Now consider $M_2 \subseteq W$ to be the optimal solution (i.e., $Z = z(M_2)$) and $M_1 = M^t$. Then, for $q \in M_2 \setminus M^t$ at iteration t, $\rho_q(M^t) \leq \rho_t, \theta \geq 0, |M^t \setminus M_2| \leq t, |M_2 \setminus M^t| \leq r$, and $z\left(M^t\right) = z(\varnothing) + \sum_{i=0}^{t-1} \rho_i$, we have:

$$
Z = z(M_2) \leq z(\varnothing) + \sum_{i=0}^{t-1} \rho_i + r\rho_t + t\theta \qquad \text{for } t = 0, \ldots, t^* - 1
$$

If $t^* < r$ and $M_1 = M^{t^*}$ then,

$$\mathrm{Z} \leq z(\varnothing) + \sum_{i=0}^{t^*-1} \rho_i + t^*\theta$$

since $\rho_{t^*} \leq 0$. □

Thus, if the greedy heuristic is applied to MUND, using $t = 0$ in (5.3) and the fact that in this case $Z^G = z(\varnothing) + \rho_0$, we obtain

$$\frac{Z - Z^G}{Z - z(\varnothing)} \leq \frac{r-1}{r}$$

Proposition 14. *If the greedy heuristic is applied to MUND, using $t = 0$ in (5.3) and the fact that in this case $Z^G = z(\varnothing) + \rho_0$, we obtain*

$$\frac{Z - Z^G}{Z - z(\varnothing)} \leq \frac{r-1}{r}$$

Proof. The inequality for $t = 0$ of (5.3) yields $Z - z(\emptyset) \leq r\rho_0 \leq r\left(Z^G - z(\emptyset)\right)$ or, equivalently,

$$\frac{Z - Z^G}{Z - z(\emptyset)} \leq \frac{r-1}{r}$$

□

A more general result for $t^* > 0$ can be obtained by using the results described above, as well as those of Lemma (4.1) and Theorem (4.1) part (a) from Nemhauser et al. (1978). Now we are going to resume those results.

Lemma 15. *See Nemhauser et al. (1978), Lemma (4.1) Let $\alpha = (r-1)/r$. Given positive integers j and r, $j < r$, and a non-negative real number b, let*

$$P(b) = rb + \min \sum_{i=0}^{L} x_i \tag{5.5}$$

$$\begin{aligned} \sum_{i=0}^{t-1} x_i + r x_t &\geq 1 - (r+t)b, \quad t = 0, \ldots, j \\ \sum_{i=0}^{j} x_i &\geq 1 - (r+j+1)b \end{aligned} \tag{5.6}$$

then

$$P(b) = \begin{cases} 1 - (j+1)b & \text{if } b \leq \alpha^{j+1}/r \\ 1 + (r-j-1)b - \alpha^{j+1} & \text{if } b \geq \alpha^{j+1}/r \end{cases}$$

and

$$\min_{b\geq 0} P(b) = 1 - \left(\frac{j+1}{r}\right)\alpha^{j+1}$$

If the last constraint is omitted from 5.6, then $P(b) = 1 + (r-j-1)b - \alpha^{j+1}$ *for all* $b \geq 0$.

Proof. The dual of problem 5.6 is

$$\begin{aligned} W(b) = rb + \max \sum_{t=0}^{i+1} \{1-(r+t)b\}u_t \\ ru_i + \sum_{t=i+1}^{j+1} u_t = 1, \quad i = 0,\dots,j \\ u_t \geq 0, \quad t = 0,\dots,j+1 \end{aligned} \tag{5.7}$$

We now are going to calculate $W(b)$, thus by LP duality $P(b)$.
Let $\lambda = 1-u_{j+1}$ in (5.7). Then we can notice that the feasible values of the remaining variables $u_t, t = 0,\dots,j$, are uniquely determined with $u_t = (\lambda/r)\alpha^{j-t}$ and

$$\sum_{t=0}^{j}\{1-(r+t)b\}u_t = \lambda\left[1-\alpha^{j+1}-(j+1)b\right]$$

Therefore

$$\begin{aligned} W(b) &= \max_{0\leq\lambda\leq 1}\left\{rb + \lambda\left[1-\alpha^{j+1}-(j+1)b\right] + (1-\lambda)[1-(r+j+1)b]\right\} \\ &= \max_{0\leq\lambda\leq 1}\left\{1-(j+1)b+\lambda\left(rb-\alpha^{i+1}\right)\right\} \end{aligned}$$

It follows immediately that $\lambda = 0\,(u_{j+1} = 1)$, if $rb < \alpha^{j+1}$, and $\lambda = 1\,(u_{j+1} = 0)$ if $rb > \alpha^{j+1}$. Therefore

$$W(b) = \max\left\{1-(j+1)b, 1+(r-j-1)b-\alpha^{j+1}\right\}$$

and

$$P(b) = W(b) = \begin{cases} 1-(j+1)b, & \text{if } b \leq \alpha^{j+1}/r \\ 1+(r-j-1)b-\alpha^{j+1}, & \text{if } b \geq \alpha^{j+1}/r \end{cases} \tag{5.8}$$

Now we observe that as $j+1 > 0$ and $r-j-1 \geq 0$

$$\min_{b\geq 0} P(b) = P\left(\frac{\alpha^{j+1}}{r}\right) = 1 - \left(\frac{j+1}{r}\right)\alpha^{j+1}$$

Consider now the case where the last constraint of (5.6) is omitted. Dropping this constraint is equivalent to finding an optimal dual solution with $u_{j+1} = 0$. But then from (5.8) we obtain $P(b) = 1 + (K-j-1)b - \alpha^{j+1}$ □

Proposition 16. *(see Nemhauser et al. (1978), Theorem 4.1). If the greedy heuristic for the MUND PR (or for the MUND BC) terminates after r^* iterations, then*

$$\frac{Z - Z^G}{Z - z(\emptyset) + r\theta} \leq \left(\frac{r^*}{r}\right) \alpha^{r^*}$$

Proof. First we consider the case $\theta < 0$. Let $z'(S) = z(S) + |S|\theta$. Then note that z' is nondecreasing, $Z - Z^G = Z' - Z'^G$ and $Z - z(\theta) + r\theta = Z' - z'(\theta)$.Therefore applying the result for $\theta = 0$ to z' yields the desired conclusion for z. For the remainder of the proof we assume that $\theta \geq 0$.

Now applying Lemma (15) with

$$b = \frac{\theta}{Z - z(\emptyset) + r\theta} \quad \text{and} \quad x_i = \frac{\rho_i}{Z - z(\emptyset) + r\theta}$$

We get from equation 5.5

$$P(b)(Z - z(\emptyset) + r\theta) \leq r\theta + \sum_{i=0}^{j} \rho_i \tag{5.9}$$

Now for $j < r^*$, $\sum_{i=0}^{i} \rho_i \leq Z^G - z(\emptyset)$ and (5.9) yields

$$P(b)(Z - z(\emptyset) + r\theta) \leq r\theta + Z^G - z(\emptyset) \tag{5.10}$$

The proof is now separated into two parts. $(r^* < r)$. Here with $j + 1 = r^*$ all of the inequalities of (5.6) are valid and from Lemma 15,

$$P(b) \geq 1 - \left(\frac{r^*}{r}\right) \alpha^{r^*} \tag{5.11}$$

By substituting (5.11) into (5.10) and doing some algebraic manipulation we obtain the result.

$$(1 - \left(\frac{r^*}{r}\right) \alpha^{r^*})(Z - z(\emptyset) + r\theta) \leq P(b)(Z - z(\emptyset) + r\theta) \leq r\theta + Z^G - z(\emptyset)$$

$$(Z - z(\emptyset) + r\theta) \leq (Z - z(\emptyset) + r\theta)(\frac{r^*}{r})\alpha^{r^*} + r\theta + Z^G - z(\emptyset)$$

$$Z \leq (Z - z(\emptyset) + r\theta)(\frac{r^*}{r})\alpha^{r^*} + Z^G$$

$$Z - Z^G \leq (Z - z(\emptyset) + r\theta)(\frac{r^*}{r})\alpha^{r^*}$$

$$\frac{Z - Z^G}{Z - z(\emptyset) + r\theta} \leq \left(\frac{r^*}{r}\right) \alpha^{r^*}$$

$(r^* = r)$. Here with $j+1=r$ only the first r inequalities of (5.6) are valid and from Lemma 15

$$P(b) \geq 1 + (r-r)b - \alpha^r = 1 - \alpha^r \tag{5.12}$$

Now substituting (5.12) into (5.10) and doing some algebraic manipulation yields the proposition. □

Remark 7. *Appling the Proposition 16 on the greedy heuristic for the MUND BC and for the MUND RP we conclude*

$$\frac{Z - Z^G}{Z - z(\emptyset) + r\theta} \leq \alpha^{r^*}$$

Remark 8. *If we let $r \to \infty$ and $r^* \to \infty$ (in the case that the heuristic can chose an unlimited number of path, or a "big" one) on Remark 7 then we get, since $r^* \leq r$*

$$\frac{Z - Z^G}{Z - z(\emptyset) + r\theta} \leq \left(\frac{r^*}{r}\right) \alpha^{r^*} \leq \alpha^{r^*} = \left(\frac{r^*-1}{r^*}\right)^{r^*} \xrightarrow{r^* \to \infty} \frac{1}{e}$$

Chapter 6

Conclusion

In this document, after giving a definition and a brief introduction to the MUND problem solution methods, we presented a new combinatorial representation of the MUND such that its objective function satisfies the submodularity property. This allowed us to present two heuristics for two variants of the problem, we could prove that those heuristics are polynomial approximation algorithms, for which we were able to establish a worst-case bound of $\left(\frac{r-1}{r}\right)^r$ (where r is a stop criterion) and reaching a bound of $1/e$ in the case of large instances which is more precise than the one that, for the best of our knowledge, were the most accurate one used before in this problem (see Wong (1980)). We were also able to provide, thanks to the submodular representation, a matroid structure for the MUND.

For future works, one interesting line of research is to computationally compare the efficiency and behavior of the greedy heuristic based on the new combinatorial representation and the existing ones; such as exact algorithms based on Benders decomposition (Zetina et al., 2019), heuristics as the presented in (Crainic et al., 2004) or those algorithms exposed on Gendron et al. (1999),(Yaghini and Akhavan, 2012) among others. Also, it is possible to look after the matroid nature of the MUND in order to propose new ways to solve the MUND or new approximation algorithms for the MUND.

Chapter 7

Bibliography

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice hall.

Andersen, J., Crainic, T. G., and Christiansen, M. (2009). Service network design with management and coordination of multiple fleets. *European Journal of Operational Research*, 193(2):377–389.

Atamturk, A. and Rajan, D. (2002). On splittable and unsplittable capacitated network design arc-set polyhedra. *Math. Program.*, 92:315–333.

Balakrishnan, A., Magnanti, T. L., and Mirchandani, P. (1991). The multi-network design problem.

Balakrishnan, N. and Wong, R. T. (1990). A network model for the rotating workforce scheduling problem. *Networks*, 20(1):25–42.

Bartholdi, J. J., Orlin, J. B., and Ratliff, H. D. (1980). Cyclic scheduling via integer programs with circular ones. *Operations Research*, 28(5):1074–1085.

Bazaraa, M., Jarvis, J., and Sherali, H. (2011). *Linear programming and network flows: Fourth edition*.

Billheimer, J. W. and Gray, P. (1973). Network design with fixed and variable cost elements. *Transportation Science*, 7(1):49–74.

Black, P. E. (2015).

Boffey, T. B. and Hinxman, A. I. (1979). Solving the optimal network problem. *European Journal of Operational Research*, 3:386–393.

Bruglieri, M., Ehrgott, M., Hamacher, H. W., and Maffioli, F. (2006). An annotated bibliography of combinatorial optimization problems with fixed cardinality constraints. *Discrete Applied Mathematics*, 154(9):1344–1357.

Chouman, M., Crainic, T. G., and Gendron, B. (2017). Commodity representations and cut-set-based inequalities for multicommodity capacitated fixed-charge network design. *Transportation Science*, 51(2):650–667.

Conforti, M., Cornuéjols, G., and Zambelli, G. (2014). *Integer Programming Models*. Springer International Publishing, Cham.

Cordeau, J.-F., Pasin, F., and Solomon, M. (2006). An integrated model for logistics network design. *Annals OR*, 144:59–82.

Crainic, T. G., Gendron, B., and Hernu, G. (2004). A slope scaling/lagrangean perturbation heuristic with long-term memory for multicommodity capacitated fixed-charge network design. *J. Heuristics*, 10:525–545.

Crainic, T. G. and Rousseau, J.-M. (1986). Multicommodity, multimode freight transportation: A general modeling and algorithmic framework for the service network design problem. *Transportation Research Part B-methodological*, 20:225–242.

Díaz, A., Glover, F., Ghaziri, H., González, J., Laguna, M., Moscato, P., and Tseng, F. T. (2000). Optimización heurística y redes neuronales.

Dionne, R. and Florian, M. (2006). Exact and approximate algorithms for optimal network design. *Networks*, 9:37 – 59.

Festa, P. and Resende, M. (2011). Grasp: Basic components and enhancements. *Telecommunication Systems*, 46:253–271.

Fischetti, M., Salvagnin, D., and Zanette, A. (2010). A note on the selection of benders' cuts. *Math. Program.*, 124:175–182.

Fragkos, I., Cordeau, J.-F., and Jans, R. (2017). *The multi-period multi-commodity network design problem*. CIRRELT, Centre interuniversitaire de recherche sur les réseaux d'entreprise ....

Frangioni, A. and Gorgone, E. (2014). Bundle methods for sum-functions with "easy" components: applications to multicommodity network design. *Mathematical Programming*, 145:133–161.

Gabrel, V., Knippel, A., and Minoux, M. (2003). A comparison of heuristics for the discrete cost multicommodity network optimization problem. *Journal of Heuristics*, 9(5):429–445.

Gendreau, M., Potvin, J.-Y., et al. (2010). *Handbook of metaheuristics*, volume 2. Springer.

Gendron, B., Crainic, T. G., and Frangioni, A. (1999). Multicommodity capacitated network design. In *Telecommunications network planning*, pages 1–19. Springer, Boston, MA,.

Geoffrion, A. and Graves, G. (2010). *Multicommodity Distribution System Design By Benders Decomposition*, volume 20, pages 35–61.

Ghamlouche, I., Crainic, T. G., and Gendreau, M. (2003). Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design. *Operations Research*, 51(4):655–667.

Gomory, R. E. (1958). An algorithm for integer solutions to linear programs. princeton ibm mathematics research project. *Techn. Report,(1)*.

Johnson, D. S., Lenstra, J. K., and Kan, A. R. (1978). The complexity of the network design problem. *Networks*, 8(4):279–285.

Kim, D. and Pardalos, P. M. (1999). A solution approach to the fixed charge network flow problem using a dynamic slope scaling procedure. *Operations Research Letters*, 24(4):195–203.

Korte, B. H., Vygen, J., Korte, B., and Vygen, J. (2018). *Combinatorial Optimization. Theory and Algorithms [6th ed.]*. Berlin: Springer.

Land, A. H. and Doig, A. G. (2010). An automatic method for solving discrete programming problems. In *50 Years of Integer Programming 1958-2008*, pages 105–132. Springer.

Lovász, L. (1983). Submodular functions and convexity. In Bachem, A., Korte, B., and Grötschel, M., editors, *Mathematical Programming The State of the Art*, pages 235–257. Springer Berlin Heidelberg.

Magnanti, T. L. and Wong, R. T. (1984). Network design and transportation planning: Models and algorithms. *Transportation science*, 18(1):1–55.

Melkote, S. and Daskin, M. S. (2001). An integrated model of facility location and transportation network design. *Transportation Research Part A: Policy and Practice*, 35(6):515–538.

Nemhauser, G. L. and Wolsey, L. A. (1988). *Integer and Combinatorial Optimization*. Wiley, New York,, USA.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294.

Ortega, F. and Wolsey, L. A. (2003). A branch-and-cut algorithm for the single-commodity, uncapacitated, fixed-charge network flow problem. *Networks: An International Journal*, 41(3):143–158.

Ortiz-Astorquiza, C., Contreras, I., and Laporte, G. (2015). The minimum flow cost hamiltonian cycle problem: A comparison of formulations. *Discrete Applied Mathematics*, 187:140–154.

Ortiz-Astorquiza, C., Contreras, I., and Laporte, G. (2017). Formulations and approximation algorithms for multilevel uncapacitated facility location. *INFORMS journal on computing*, 29(4):767–779.

Paraskevopoulos, D. C., Bektaş, T., Crainic, T. G., and Potts, C. N. (2016). A cycle-based evolutionary algorithm for the fixed-charge capacitated multi-commodity network design problem. *European Journal of Operational Research*, 253(2):265–279.

Raack, C., Koster, A. M., Orlowski, S., and Wessäly, R. (2011). On cut-based inequalities for capacitated network design polyhedra. *Networks*, 57(2):141–156.

Randazzo, C. and Luna, H. P. L. (2001). A comparison of optimal methods for local access uncapacitated network design. *Annals of Operations Research*, 106(1):263–286.

Scott, A. J. (1969). The optimal network problem: Some computational procedures. *Transportation Research/UK/*, 3(2).

Vondrák, J. (2007). Submodularity in combinatorial optimization.

Wolsey, L. (2020). *Integer Programming.* John Wiley & Sons, USA.

Wong, R. T. (1980). Worst-case analysis of network design problem heuristics. *SIAM Journal on Algebraic Discrete Methods*, 1(1):51–63.

Yaghini, M. and Akhavan, R. (2012). Multicommodity network design problem in rail freight transportation planning. *Procedia - Social and Behavioral Sciences*, 43:728–739. 8th International Conference on Traffic and Transportation Studies (ICTTS 2012).

Zetina, A. C., Cordeau, J.-F., and Contreras, I. (2017). *Exact algorithms for the multicommodity uncapacitated fixed-charge network design problem.* CIRRELT, Centre interuniversitaire de recherche sur les réseaux d'entreprise ....

Zetina, C. A., Contreras, I., and Cordeau, J.-F. (2019). Exact algorithms based on benders decomposition for multicommodity uncapacitated fixed-charge network design. *Computers & Operations Research*, 111:311–324.